

Scilab on Multi-Cores, Clusters & Clouds with ProActive Parallel Suite

D. Caromel, F. Viale, et al.

Agenda

1. Background
2. ProActive Parallel Suite:
Programming, Optimizing
Scheduling
3. Scilab bindings



Key Objectives

- ❑ **Parallel Programming Model and Tools**
 - desperately needed
 - for the masses
 - for new architectures (Multi-cores)
- ❑ **As Effective as possible:**
 - Efficient
 - However Programmer Productivity is first Key
- ❑ **For both Multi-cores and Distributed**
 - Actually the way around
- ❑ **Some Handling of ``Large-scale'' (Grid, Clouds)**



1. Background

- ❑ A joint team, Now about 35 persons
- ❑ 2004: First ProActive User Group – Grid Plugtest
- ❑ 2009, April: ProActive 4.1, Distributed & Parallel:
From Multi-cores to Enterprise GRIDs



OASIS Team Composition (35)

□ Researchers (5):

- D. Caromel (UNSA, Det. INRIA)
- E. Madelaine (INRIA)
- F. Baude (UNSA)
- F. Huet (UNSA)
- L. Henrio (CNRS)

□ PhDs (11):

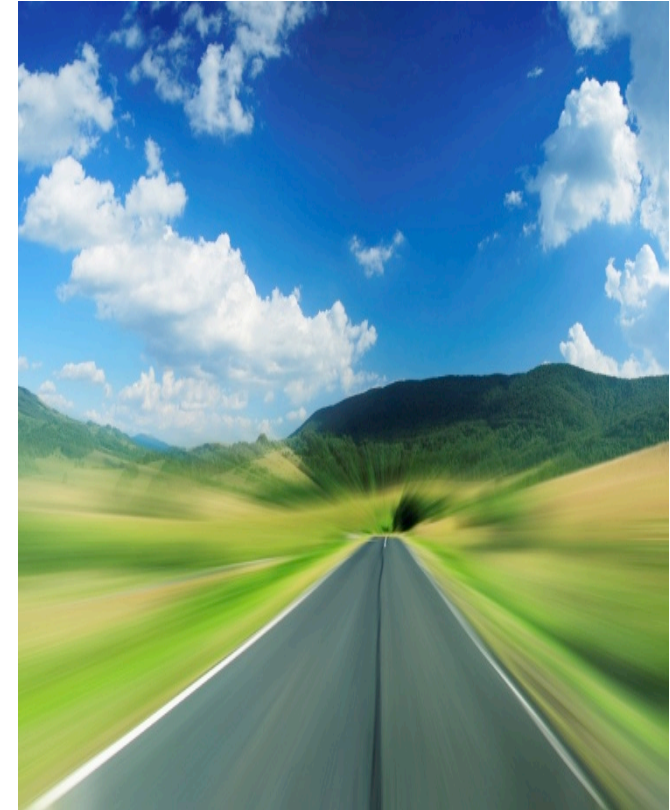
- Antonio Cansado (INRIA, Conic)
- Brian Amedro (SCS-Agos)
- Cristian Ruz (INRIA, Conicyt)
- Elton Mathias (INRIA-Cordi)
- Imen Filali (SCS-Agos / FP7 SC)
- Marcela Rivera (INRIA, Conicyt)
- Muhammad Khan (STIC-Asia)
- Paul Naoumenko (INRIA/Région)
- Viet Dung Doan (FP6 Bionets)
- Virginie Contes (SOA4ALL)
- Guilherme Pezzi (AGOS, CIFR)

□ + Visitors + Interns



Located in Sophia Antipolis, between
Nice and Cannes,
Visitors and Students Welcome!

Startup Company Born of INRIA



- ❑ Co-developing, Support for [ProActive Parallel Suite](#)
- ❑ Worldwide Customers: Fr, UK, Boston USA



2. ProActive Parallel Suite



Parallel Acceleration Toolkit in Java:

Parallelism:

Multi-Core+Distributed



Open Source Used in production by industry



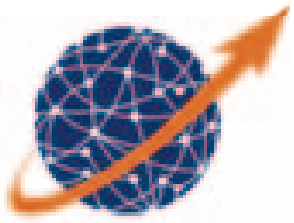
OW2: Object Web + Orient Ware



Leading Open Source Middleware



ProActive Parallel Suite



PROGRAMMING

Java Parallel Frameworks

for HPC, Multi-Cores, Distribution, Enterprise Grids and Clouds.

Featuring: Async. comms, Master-Worker, Monte-Carlo, SPMD, components and legacy code wrapping.

OPTIMIZING

Eclipse GUI (IC2D)

for Developing, Debugging, Optimizing your parallel applications.

Featuring: graphical monitoring and benchmarking with report generation.

SCHEDULING

Multi-Language Scheduler

for Workflows made of C, C++, Java, Scripts, Matlab, Scilab tasks.

Featuring: graphical user interface, resource acquisition and virtualization.

ProActive Contributors

Abhijeet Gaikwad
(Option Pricing)
Abhishek-Rajeev Gupta
Antonio Cansado
Baptiste De Stefano
Bastien Sauvan
Brian Amedro (SPMD)
Cédric Dalmasso (Component)
Clement Mathieu (Core,
GCM Deployment)
Elaine Isnard
Elton Mathias
Eric Madelaine
Etienne Vallette-De-Osia
Fabien Viale (Matlab, Scilab)
Fabrice Huet (Mobility, P2P)
Florin Bratu
Franca Perrina
Francoise Baude
Germain Sigety (Scheduling)
Guillaume Laurent
Guilherme Perretti Pezzi
Imen Filiali
Jean-Luc Scheefer (Scheduling)

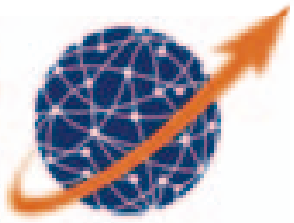
Jean-Michael Guillamume
Johann Fradj (Scheduling)
Jonathan Martin
Julian Krzeminski
Kamran Qadir
Khan Muhammad
Laurent Vanni
Ludovic Henrio
Marcela Rivera
Mario Leyton (Skeleton)
Maxime Menant
Nicolas Dodelin
Olivier Helin
Paul Naoumenko
Regis Gascon
Tomasz Dobek
Vasile Jureschi (Technical Writer)
Viet Dong Doan
Vincent Cave (Legacy Wrapping)
Virginie Contes (OSGi, WS)
Yu Feng
Yulai Yuan
Zhihui Dai

Arnaud Contes (Core, Security, Debugging)
Christian Delbé (FT, Scheduling)
Emil Salageanu (IC2D, Agent)
Vladimir Bodnartchouk (IC2D, TimIt)

Alexandre di
Costanzo (P2P, B&B)
Boutheina Bennour
Guillaume Chazarain (DGC)
Julien Vayssiere
(MOP, Active Objects)

Lionel Mestre
Laurent Baduel (Group
Communications)
Matthieu Morel (Initial
Component Work)
Nadia Ranaldo
(Core, Deployment)
Romain Quilici

ProActive Parallel Suite



PROGRAMMING

Java Parallel Frameworks

for HPC, Multi-Cores, Distribution, Enterprise Grids and Clouds.

Featuring: Async. comms, Master-Worker, Monte-Carlo, SPMD, components and legacy code wrapping.

OPTIMIZING

Eclipse GUI (IC2D)

for Developing, Debugging, Optimizing your parallel applications.

Featuring: graphical monitoring and benchmarking with report generation.

SCHEDULING

Multi-Language Scheduler

for Workflows made of C, C++, Java, Scripts, Matlab, Scilab tasks.

Featuring: graphical user interface, resource acquisition and virtualization.

ProActive Parallel Suite



PROGRAMMING

Java Parallel Frameworks

for HPC, Multi-Cores,
Distribution, Enterprise
Grids and Clouds.

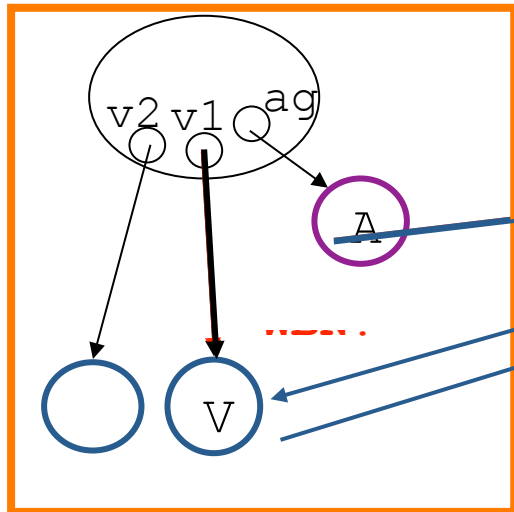
Featuring: Async. comms,
Master-Worker, Monte-
Carlo, SPMD, components
and legacy code wrapping.

ProActive : Active objects

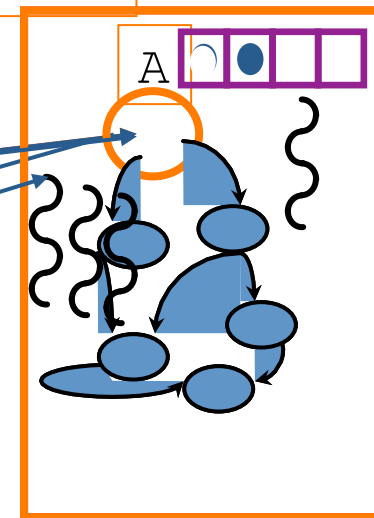
```

● A ag = newActive ("A", [...], VirtualNode)
● V v1 = ag.foo (param);
● V v2 = ag.bar (param);
...
● v1.bar(); //Wait-By-Necessity
    
```

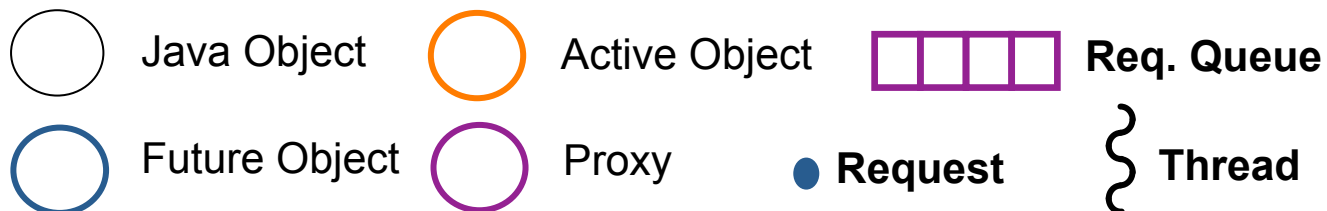
JVM



JVM



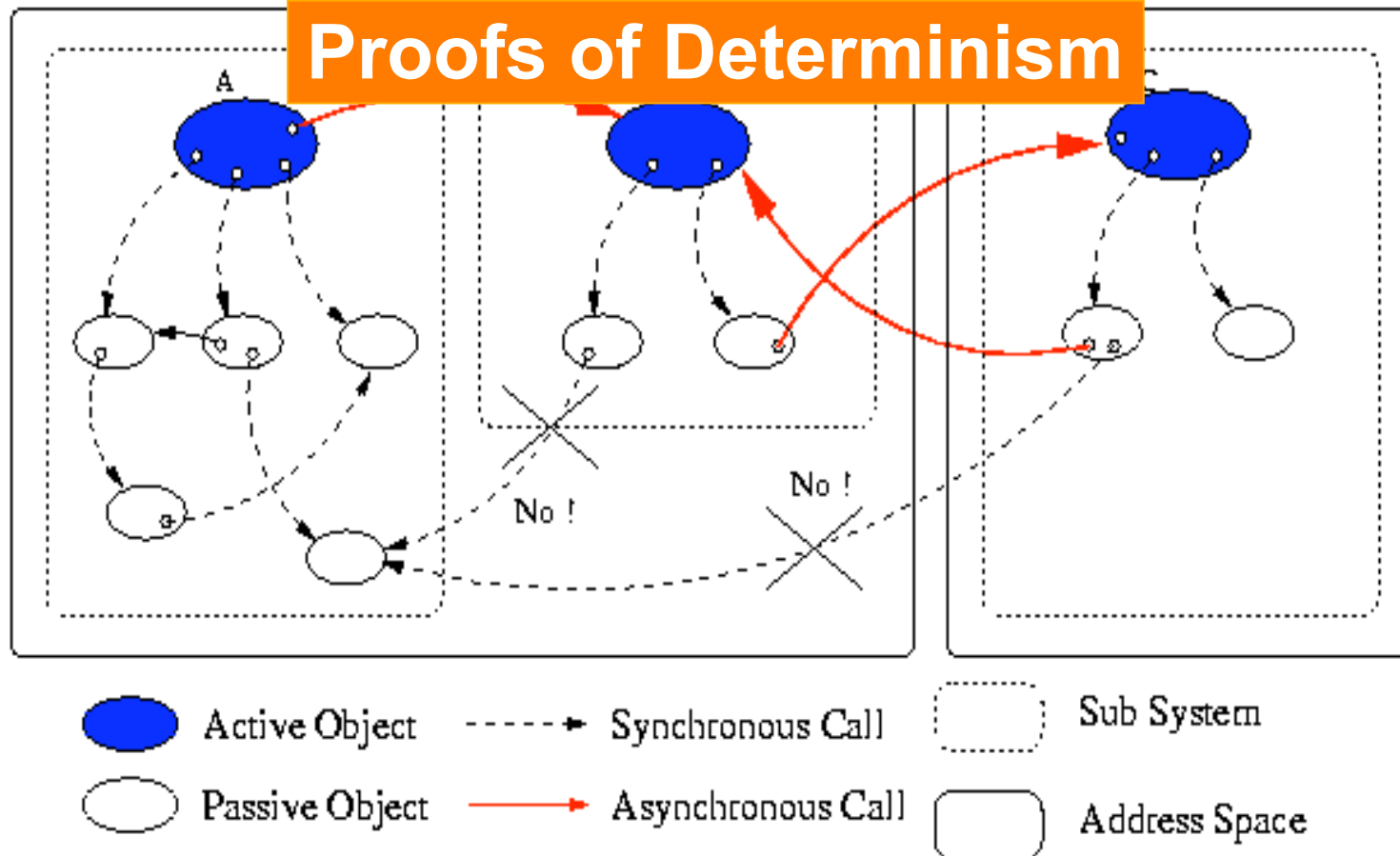
Wait-By-Necessity
is a
Dataflow
Synchronization



Standard system at Runtime: No Sharing

NoC: Network On Chip

Proofs of Determinism

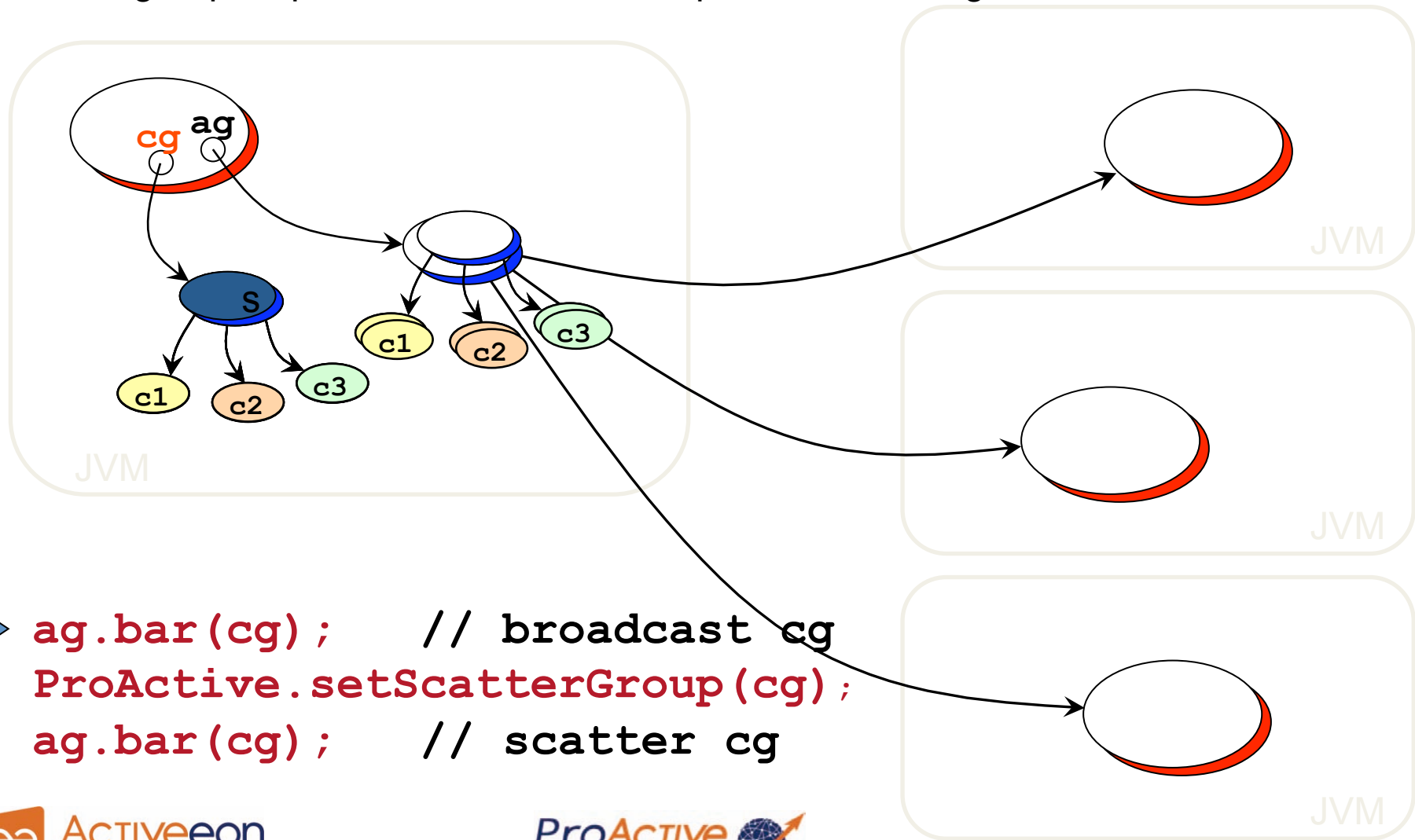


TYPED ASYNCHRONOUS GROUPS

Broadcast and Scatter

Broadcast is the default behavior

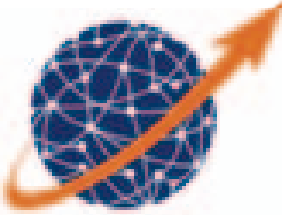
Use a group as parameter, Scattered depends on rankings



➔ `ag.bar (cg) ; // broadcast cg`
`ProActive.setScatterGroup (cg) ;`
`ag.bar (cg) ; // scatter cg`

Optimizing

ProActive Parallel Suite



PROGRAMMING

Java Parallel Frameworks

for HPC, Multi-Cores, Distribution, Enterprise Grids and Clouds.

Featuring: Async. comms, Master-Worker, Monte-Carlo, SPMD, components and legacy code wrapping.

OPTIMIZING

Eclipse GUI (IC2D)

for Developing, Debugging, Optimizing your parallel applications.

Featuring: graphical monitoring and benchmarking with report generation.

Monitoring - Eclipse SDK

File Edit Navigate Search Project Run Control Monitoring Window Help

Monitoring

Virtual nodes

☒ Renderer ☐ DefaultVN ☐ Dispatcher ☐ User

bebita.inria.fr:1099:OS u...

PA_JVM1357457629_be...

Node Node60562498...

DinnerLayout#2

Table#3

Philosopher#4

Philosopher#5

Philosopher#6

Philosopher#7

Philosopher#8

PA_JVM-436155261_be...

Node Renderer-127...

C3DRendering...

PA_JVM-1672076495_b...

Node Dispatcher-5...

C3DDispatche...

PA_JVM-294719007_be...

Node User1602644...

C3DUser13

PA_JVM-1631909824_b...

Node Renderer1307...

C3DRendering...

duff.inria.fr:1099:OS und...

PA_JVM1530781642_du...

Node Renderer1174...

C3DRendering...

sidonie.inria.fr:1099:OS ...

PA_JVM-772843461_sl...

Node Renderer-151...

C3DRendering...

PA_JVM-60562498_sl...

Node Node-4551863...

☒ Display topology ☐ Proportional ☐ Ratio ☒ Filaire Reset Topology ☒ Monitoring enable

Console

Monitoring

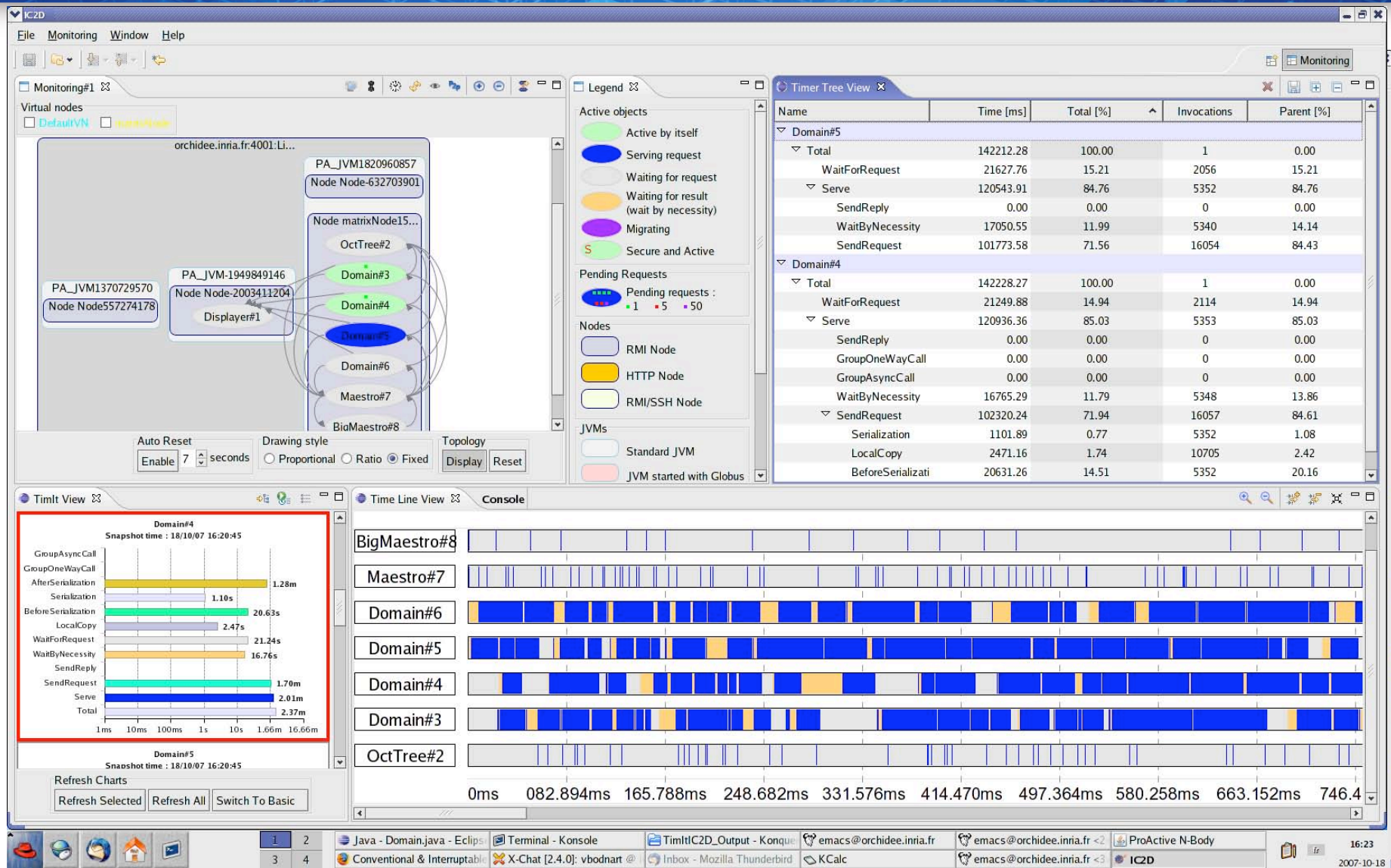
15:09:15 => NodeObject id=Node-455186381 already monitored, check for new active objects

Legend

Job Monitoring View

- DefaultVN (JOB-1357457629)
 - bebita.inria.fr:1099:OS un
 - PA_JVM1357457629_
 - Node Node60562498
 - DinnerLayout#2
 - Table#3 (JOB-1357457629)
 - Philosopher#4 (JOB-1357457629)
 - Philosopher#5 (JOB-1357457629)
 - Philosopher#6 (JOB-1357457629)
 - Philosopher#7 (JOB-1357457629)
 - Philosopher#8 (JOB-1357457629)
 - sidonie.inria.fr:1099:OS u
 - Dispatcher (JOB-1672076495)
 - User (JOB-294719007)
 - bebita.inria.fr:1099:OS un
 - PA_JVM-294719007_
 - Node User1602644
 - C3DUser13 (JOB-294719007)
 - Renderer (JOB-1672076495)
 - bebita.inria.fr:1099:OS un
 - PA_JVM-1631909824_
 - Node Renderer1307
 - C3DRendering...

IC2D

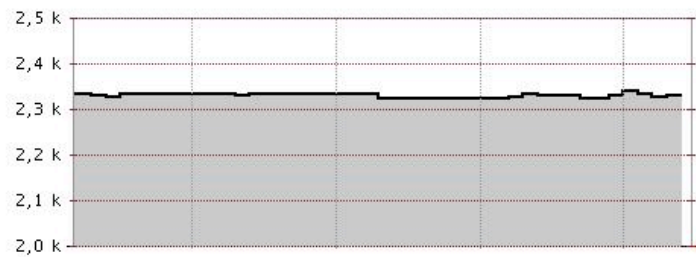


ChartIt

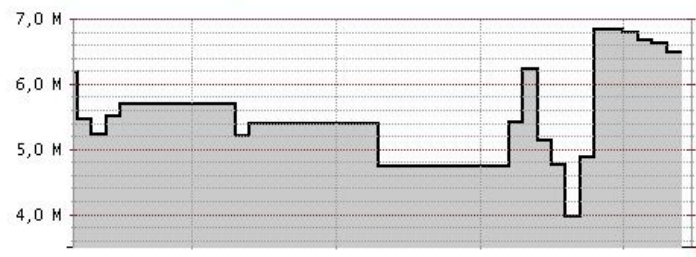
PA_JVM251111462

Charts

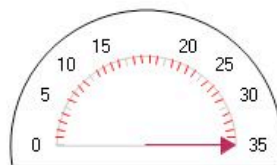
▼ Chart#1 [LoadedClassCount]



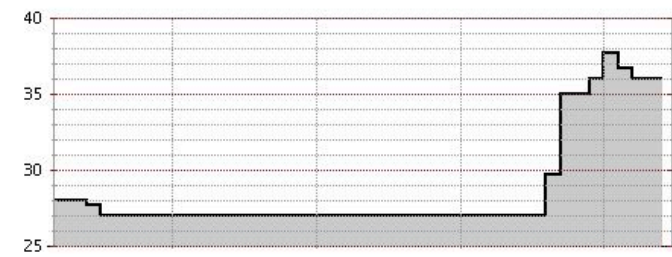
▼ Chart#2 [UsedHeapMemory]



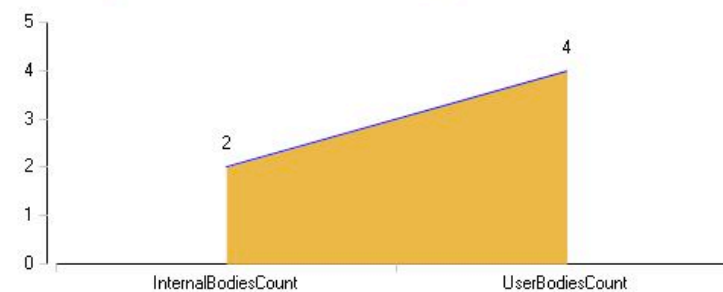
▼ Chart#5 [ThreadCount] Meter



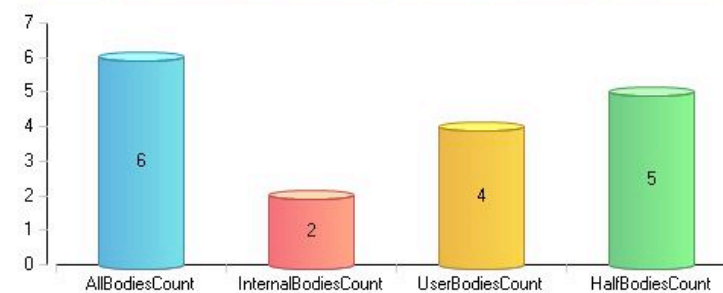
▼ Chart#3 [ThreadCount]



▼ Chart#4 [InternalBodiesCount, UserBodiesCount]



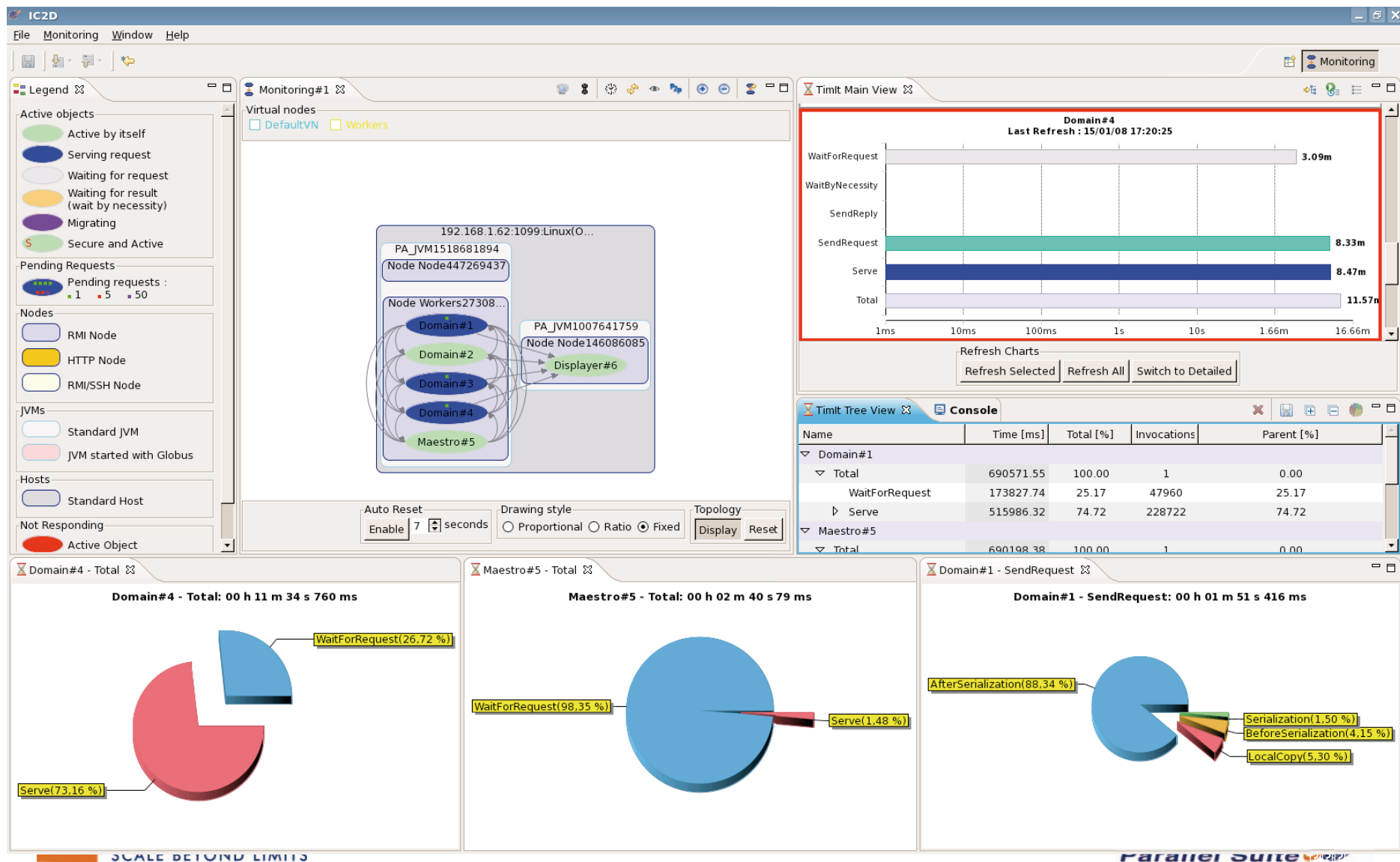
▼ Chart#6 [AllBodiesCount, InternalBodiesCount, UserBodiesCount, HalfBodiesCount]



SCALE BEYOND LIMITS



Pies for Analysis and Optimization



Scheduling

ProActive Parallel Suite

PROGRAMMING

Java Parallel Frameworks

for HPC, Multi-Cores, Distribution, Enterprise Grids and Clouds.

Featuring: Async. comms, Master-Worker, Monte-Carlo, SPMD, components and legacy code wrapping.

OPTIMIZING

Eclipse GUI (IC2D)

for Developing, Debugging, Optimizing your parallel applications.

Featuring: graphical monitoring and benchmarking with report generation.

SCHEDULING

Multi-Language Scheduler

for Workflows made of C, C++, Java, Scripts, Matlab, Scilab tasks.

Featuring: graphical user interface, resource acquisition and virtualization.

Scheduler: User Interface

ProActive Scheduler

File Window Help

Scheduler

Jobs

Pending (8)

Id	State	User	Priority	Name
172	Pending	user1	Low	job_2_tasks
173	Pending	user1	Low	job_2_tasks
174	Pending	user1	Low	job_2_tasks
176	Pending	user1	Low	job_2_tasks
177	Pending	user1	Low	job_2_tasks
178	Pending	user1	Low	job_2_tasks
179	Pending	user1	Low	job_2_tasks
180	Pending	user1	Low	job_2_tasks

Running (13)

Id	State	Progress	# Finishe	User	Priority	Nam
54	Running		1/2	user1	Low	job_2_t
55	Running		0/2	user1	Low	job_2_t
56	Running		1/2	user1	Low	job_2_t
160	Running		1/2	user1	Low	job_2_t
161	Running		1/2	user1	Low	job_2_t
162	Running		1/2	user1	Low	job_2_t
163	Running		1/2	user1	Low	job_2_t
164	Running		1/2	user1	Low	job_2_t
165	Running		1/2	user1	Low	job_2_t
166	Running		1/2	user1	Low	job_2_t
168	Running		0/2	user1	Low	job_2_t
169	Running		0/2	user1	Low	job_2_t
170	Running		0/2	user1	Low	job_2_t

Finished (11)

Id	State	User	Priority	Name
152	Finished	user1	Low	job_2_tasks
167	Finished	user1	Normal	job_2_tasks
171	Finished	user1	Normal	job_2_tasks
153	Finished	user1	Low	job_2_tasks
175	Finished	user1	Normal	job_2_tasks
154	Finished	user1	Low	job_2_tasks
155	Finished	user1	Low	job_2_tasks
156	Finished	user1	Low	job_2_tasks
157	Finished	user1	Low	job_2_tasks
158	Finished	user1	Low	job_2_tasks
159	Finished	user1	Low	job_2_tasks

STARTED

Console Tasks Users

Job 55 has 2 tasks

Id	State	Name	Host name	Start time	Finished time	Re-run	Description
55000: Running	task1	eon8.inria.fr	16:09:28 08/27/08	Not yet	0/3	task WaitAndPrint - will sleep for 3s	
55000: Running	task2	eon8.inria.fr	16:09:28 08/27/08	Not yet	0/1	task WaitAndPrint - will sleep for 20s	

Job Info Result Preview

Property	Value
Id	55
State	Running
Name	job_2_tasks
Priority	Low
Pending tasks number	0
Running tasks number	2
Finished tasks number	0
Total tasks number	2
Submitted time	16:09:28 08/27/08

3. Scilab <-> ProActive

Scilab <-> ProActive

- ProActive Scheduler => Scilab
 - Scilab tasks

- Scilab => ProActive Scheduler

Scilab Tasks

```
<task name="t1">
  <description>
    Calculates the numerical integration of  $\sin(x)$  between
     $i \cdot \pi/n$  and  $(i+1) \cdot \pi/n$ 
  </description>
  <selection>
    <script>
      <file
        url="http://proactive.inria.fr/userfiles/file/scripts/checkScilab.js" />
      </script>
    </selection>
    <javaExecutable
      class="org.ow2.proactive.scheduler.ext.scilab.ScilabTask">
        <parameters>
          <parameter name="scriptFile"
            value="{SCILAB_SCRIPTS}/intsin.sci" />
          <parameter name="input" value="i=0;n=5;" />
          <parameter name="outputs" value="out" />
        </parameters>
      </javaExecutable>
    </task>
```


Scilab => Scheduler

❑ Scilab's toolbox PAScheduler

- Installation : builder.sce, loader.sce scripts
- Prerequisite : Scilab 5.1

❑ Usage :

- PAconnect
- PASolve

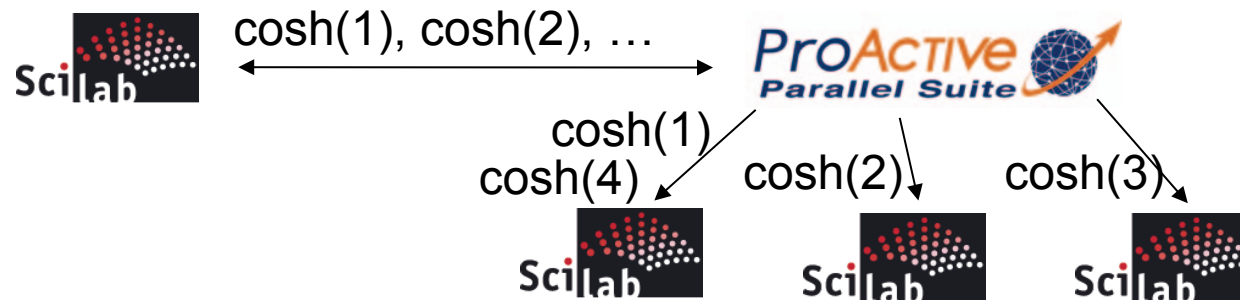
Scilab => Scheduler

```
❑ >> PAconnect('rmi://bison:1099')
```



```
❑ >> res = PAsolve('cosh', list(1,2,3,4,5));
```

- Wait...



Scheduler: User Interface

The screenshot displays the ProActive Scheduler User Interface. The main window is titled "ProActive Scheduler" and contains three panels: "Pending (8)", "Running (13)", and "Finished (11)". Each panel shows a table of jobs with columns for Id, State, User, Priority, and Name. The "Running" panel also includes a "Progress" column with a visual progress bar and a "# Finishe" column. Below the job lists is a "STARTED" button. At the bottom, there is a "Console" panel showing a message "Job 55 has 2 tasks" and a table of tasks with columns for Id, State, Name, Host name, Start time, Finished time, Re-run, and Description. To the right of the console is a "Job Info" panel showing details for job 55, including its state, name, priority, and task counts.

Pending (8)				
Id	State	User	Priority	Name
172	Pending	user1	Low	job_2_tasks
173	Pending	user1	Low	job_2_tasks
174	Pending	user1	Low	job_2_tasks
176	Pending	user1	Low	job_2_tasks
177	Pending	user1	Low	job_2_tasks
178	Pending	user1	Low	job_2_tasks
179	Pending	user1	Low	job_2_tasks
180	Pending	user1	Low	job_2_tasks

Running (13)							
Id	State	Progress	# Finishe	User	Priority	Name	
54	Running		1/2	user1	Low	job_2_t	
55	Running		0/2	user1	Low	job_2_t	
56	Running		1/2	user1	Low	job_2_t	
160	Running		1/2	user1	Low	job_2_t	
161	Running		1/2	user1	Low	job_2_t	
162	Running		1/2	user1	Low	job_2_t	
163	Running		1/2	user1	Low	job_2_t	
164	Running		1/2	user1	Low	job_2_t	
165	Running		1/2	user1	Low	job_2_t	
166	Running		1/2	user1	Low	job_2_t	
168	Running		0/2	user1	Low	job_2_t	
169	Running		0/2	user1	Low	job_2_t	
170	Running		0/2	user1	Low	job_2_t	

Finished (11)				
Id	State	User	Priority	Name
152	Finished	user1	Low	job_2_tasks
167	Finished	user1	Normal	job_2_tasks
171	Finished	user1	Normal	job_2_tasks
153	Finished	user1	Low	job_2_tasks
175	Finished	user1	Normal	job_2_tasks
154	Finished	user1	Low	job_2_tasks
155	Finished	user1	Low	job_2_tasks
156	Finished	user1	Low	job_2_tasks
157	Finished	user1	Low	job_2_tasks
158	Finished	user1	Low	job_2_tasks
159	Finished	user1	Low	job_2_tasks

STARTED

Console

Job 55 has 2 tasks

Id	State	Name	Host name	Start time	Finished time	Re-run	Description
55000	Running	task1	eon8.inria.fr	16:09:28 08/27/08	Not yet	0/3	task WaitAndPrint - will sleep for 3s
55000	Running	task2	eon8.inria.fr	16:09:28 08/27/08	Not yet	0/1	task WaitAndPrint - will sleep for 20s

Job Info

Property	Value
Id	55
State	Running
Name	job_2_tasks
Priority	Low
Pending tasks number	0
Running tasks number	2
Finished tasks number	0
Total tasks number	2
Submitted time	16:09:28 08/27/08

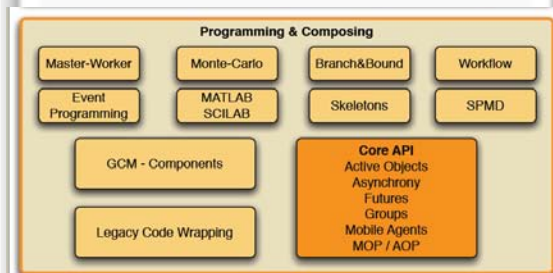
Summary

Conclusion: Currently Available



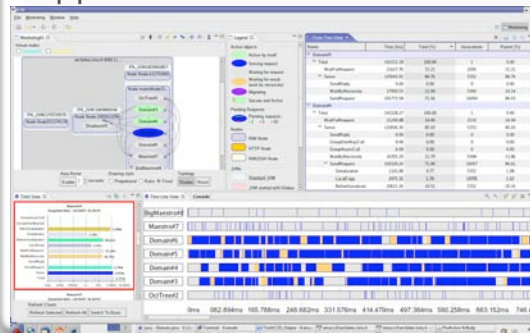
PROGRAMMING

Java Parallel Frameworks
for HPC, Multi-Cores,
Distribution, Enterprise
Grids and Clouds.



OPTIMIZING

Eclipse GUI (IC2D)
for Developing, Debugging,
Optimizing your parallel
applications.



SCHEDULING

Multi-Language Scheduler
for Workflows made of C,
C++, Java, Scripts, Matlab,
Scilab tasks.

A screenshot of the ProActive Scheduler interface showing a table of jobs. The table has columns for Job ID, State, User, Priority, Name, Progress, # Finishes, User, Priority, and Name. The jobs are listed in two sections: "Pending (8)" and "Running (13)".

Job ID	State	User	Priority	Name	Progress	# Finishes	User	Priority	Name
172	Pending	user1	Low	job_2_tasks					
173	Pending	user1	Low	job_2_tasks					
174	Pending	user1	Low	job_2_tasks					
176	Pending	user1	Low	job_2_tasks					
177	Pending	user1	Low	job_2_tasks					
178	Pending	user1	Low	job_2_tasks					
179	Pending	user1	Low	job_2_tasks					
180	Pending	user1	Low	job_2_tasks					
54	Running					1/2	user1	Low	job_2_1
55	Running					1/2	user1	Low	job_2_1
56	Running					1/2	user1	Low	job_2_1
160	Running					1/2	user1	Low	job_2_1
161	Running					1/2	user1	Low	job_2_1
162	Running					1/2	user1	Low	job_2_1
163	Running					1/2	user1	Low	job_2_1
164	Running					1/2	user1	Low	job_2_1
165	Running					1/2	user1	Low	job_2_1
166	Running					1/2	user1	Low	job_2_1
168	Running					0/2	user1	Low	job_2_1
169	Running					0/2	user1	Low	job_2_1
170	Running					0/2	user1	Low	job_2_1

**Further into the direction of:
Multi-Core + Distributed**